Annales UMCS Informatica AI 8(1) (2008) 75-86
10.2478/v10065-008-0008-2

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Static methods for object reconstruction overview: for medical diagnosis

Karolina Nurzyńska*

Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

Abstract

This article presents the overview of static methods exploited for object reconstruction from point cloud or the special case which are the sets of parallel contours gathered from the medical scanners. It includes a brief description of each method and a comparison of their performance in respect to the achieved object appearance, an impact of noisy data, possible types of object reconstruction and time consumption. The aim of this comparison is to find which of the presented methods are promising for object reconstruction needed for medical diagnosis.

1. Introduction

Due to fast technology development it not only has a greater application in everyday life but also becomes a source of knowledge. Thanks to it, we can frequently see invisible things, which were impossible to observe with the naked eye or look into inaccessible places. It gives scientists a powerful tool for further research and development.

One of the technologies evolved in the second half of the twentieth century is the one utilized for developing scanner machines allowing 3D data acquisition. The very beginning of research might be acknowledged as the elaboration of the first computer tomography (CT) scanner by Hounsfield in the Thorn EMI Central Research laboratories in 1967, which was utilized for public usage five years later. In succeeding years the development allowed scan quality improvement and speed acceleration. The result of CT scanner operation is the image of an inner object structure; hence it finds a broad application from medical diagnosis through archaeological research to computer-aided design (CAD) domain. Different scanners, also designed for medical diagnosis, are those based on emission tomography like single photon emission computer

* *e-mail address*: Karolina.Nurzynska@polsl.pl

tomography (SPECT) and positron emission tomography (PET). Its work focuses on the visualisation of tissues function in living organisms, which emit different quantities of radioactive substances depending on activity their life activities. It is also possible to come across 3D scanners which enable an external object surface acquisition used, among others, in the creation of virtual museums.

The main aim of this work is to give a brief review of static methods developed for 3D object reconstruction from data acquired from scanners. There exist also dynamic methods utilized for object reconstruction; however, this article does not deal with them.

The article is organized as follows. In Section 2 the basic concepts are introduced and methods requirement specification is outlined. The description of leading algorithms is presented in Section 3, while Section 4 compares them according to the chosen features enumerated in Section 2. Section 5 includes conclusions and further work directions.

2. Method features

The scanner work results in the set of parallel cross-sections through the scanned object. Usually the slices are done in the plane perpendicular to the longest dimension of the object (*Z* axis). Additionally, the resolution in the *Z* axis, that is the cross-sections count, is several-times smaller than in the *XY* plane. One manner of such data visualization is to locate them in the space as a pile of 2D images. This solution is easy to obtain, however, often not sufficient for correct and simple 3D visualization. In connection with this problem need for 3D object reconstruction and visualization algorithms arises.

The main task for reconstruction methods is to convert the input data into the 3D irregular polygonal mesh, which can be easily displayed. The input information can be given as a point cloud, which is a set of three-dimensional points describing the outlines or surface features of an object. However, it is also possible to extract the contours, which describe the object outline on each cross-section. This is a special case of the point cloud, since here additional information about the connection between some points is stored and can be further utilized (see Fig. 1).

The reconstructed object and the process of reconstruction itself should fulfil some requirements:

– bring the plausible object appearance,
– result in the irregular polygonal mesh,
– manage with closed and opened objects surface,
– manage with branching structures,

– not create holes in the object surface,
– work in a limited time.

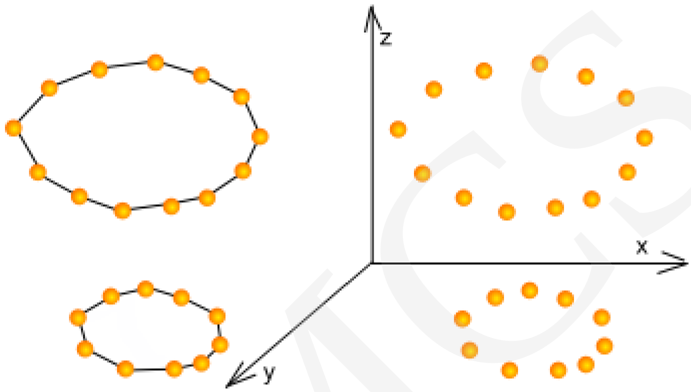

Fig. 1. Left side – contour. Right side – point cloud

When reconstructing an object from the data acquired from the scanner, it is difficult to determine any specific formula which could reflect the reconstruction quality. The problem arises from two factors. Firstly, the original object is unknown, as the only information we possess is tomography data. Secondly, even if we knew the original object, it would be difficult to suggest any mathematical formula reflecting the reconstruction quality. Therefore in this problem this parameter is subjective and depends, among others, on special needs of the system.

For visualization the irregular polygonal mesh representation of the object was chosen. This solution allows easy description of both detailed and simple object surfaces. It is widely spread as an input data format in 3D object visualization environments. Obviously, the parametric surface representation of the object is more compact; however, it provides many difficulties while creating the surface from the point cloud especially when the object surface is not continuous and the joint edges must be found. A similar problem arises when using alpha shapes defined by Edelsbrunner and Mücke in [1].

The chosen method should be able to reconstruct any shape. Therefore the problem of open and closed objects, holes, and branching should be addressed. Two-dimensional manifold without boundary in three-dimensional space is understood as a closed object while the open surface is two-dimensional manifold with the boundary. In both previous cases the object may branch. Finally, the reconstruction method should not create any additional holes, which usually results from algorithms imperfection.

The evaluation time of the reconstruction method will play a crucial role in the real-time and soft real-time systems. However, in the case of offline object reconstruction is of minor importance meaning. Therefore, the suitability of the method depends on subjective system requirements.

3. Methods overview

This section includes a brief review of static methods for an object reconstruction. Those methods in the iterative manner reconstruct the irregular polygonal mesh from the point cloud. There could be distinguished explicit methods – which base on object geometry – and implicit methods – which use scalar functions. During the performance all input data must be looked through and classified.

3.1. Marching cubes

The marching cubes method was introduced by Lorensen and Cline [2]. In the original version it allowed to achieve an irregular polygonal mesh which was an approximation of a given surface. For the input data – the surface later on the volumetric data – the surrounding cube is created, which is divided into smaller ones. The number of divisions resembles the resolution of the resulting mesh. However, as easily noticeable, the higher resolution, the longer processing time as each cube must be classified into one of the groups from the lookup table (see Fig. 3). Classification consists in specification which from the cube vertices is inside the surface (value 1) or outside (value 0). Hence it is clear that the cube edges are cut by the surface. As a cube has 8 vertices and two create an edge, there are 256 possibilities of surface triangulation. However, thanks to the symmetry, the number can be reduced to 15 classes which are depicted in Fig. 2.

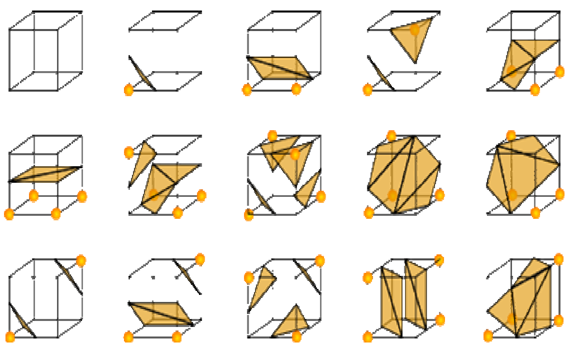

Fig. 2. The marching cubes triangulation classes used in the marching cubes algorithm. The marked vertices respond to vertices which are inside the object

Bourke [3] expanded this method for the input data given as point cloud or three-dimension scalar field. In this case we lack the information about the surface cross-cut through the cube edges; however, the scalar field value comparison allows replacing this information. Subsequently, Bourke [4] introduced a method which divided the space into regular tetrahedrons and exploited them for the irregular polygonal mesh creation.

The marching cubes method easily triangulates the object surface and allows achieving high mesh resolution. But 15 triangulation classes proved not to be sufficient for proper (without holes and ambiguity) object reconstruction. Therefore, there have been added new triangulation classes by Chernyaev [5] and ambiguity elimination [6].

On the contour data retrieved from the human body cross-sections gathered from CT scanners, this method proved to reconstruct a surface with a "staircase effect" [7,8]. It is due to the distance between subsequent slices. Therefore in the first step the lacking data is approximated with the Eulerian methods (see point 3.5) and afterwards the marching cube algorithm is executed.

3.2. Marching voxels

The marching voxels method [9] addresses the problems which are troublesome for marching cubes: ambiguity, holes and evaluation time. The authors suggest that all these problems are due to division of the space into cubes and therefore their solution uses the vertices for algorithm work. It processes the input data in three stages. Firstly, there are created triangles among all vertices, which protects from holes generation. Secondly, those triangles are joined in polygons by adjoining edges, which creates a polygonal mesh. Finally, for visualization, the data is projected on the 2D plane.

3.3. Delaunay triangulation

The Delaunay triangulation method is based on the observation by Delaunay [10], which states that for a set of points *P* there exists such a triangulation *DT(P)* that no point in *P* is inside the circumference of any triangle in *DT(P)*. The Delaunay triangulation for a discrete point set is unequivocal with the Voronoi diagram.

There are some algorithms implementing this triangulation, for instance those introduced by Yuan and Fitzimons [11], Lee [12] or Si [13]. The algorithm presented by Yuan and Fitzimons firstly creates a triangle that all points from *P* are inside it. Next, in the iterative manner, in each step one point is added. As the new point does not fulfil the assumption – is in the circumference built on the main triangle – the flip operation takes place. It searches for all triangles in which circumference is the point, removes the edges of these triangles, and in their places adds the connection with the added point as depicted in Fig. 3.

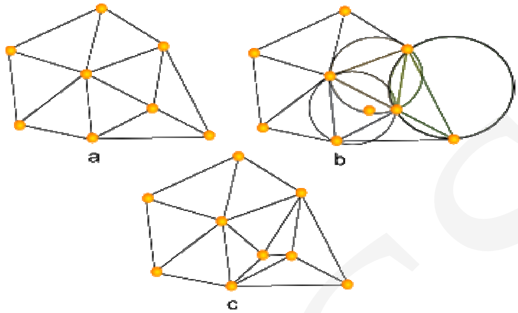

Fig. 3. Delaunay triangulation – example of inserting new vertex into the mesh. a. the mesh, b. the mesh with drawn circumferences for new vertex, c. flipped edges and final mesh fulfilling the Delaunay assumption

3.4. Contour stitching

The contour stitching, known also as a Langrangian method, as an input data needs a set of contours and results in the irregular polygonal mesh which interpolates the object surface. The aim of this method is to create the connections between points in the adjacent contours. For achieving satisfactory results, three issues should be addressed:

- correspondence,
- tiling,
- and branching.

The correspondence problem is defined that if we have a set of vertices in the contour *A* and a set of vertices in the contour *B*, which are to be joined, it is necessary to find the connections between them (see Fig. 4). Wrong connections result in the effect of surface twisting. To solve this problem Ganapathy & Dennehy [12] suggest choosing as a starting point the point with the smallest value of the *x* coefficient. On the other hand, Parker & Attia [15] prefer to find the global length of the edges needed for connections and then choose the shortest solution.

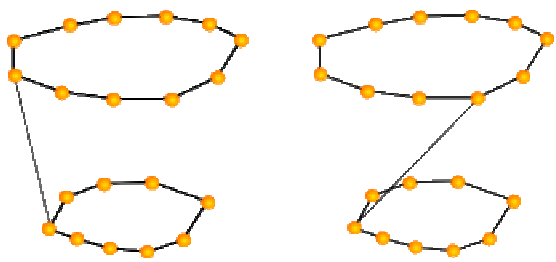

Fig 4. Contour stitching – correspondence problem. On the left side correct connection; on the right side one which could cause twisting

The tiling determines whether the points connection generates a triangle or quadrilaterals as depicted in Fig. 5. It influences the smoothness of the surface.

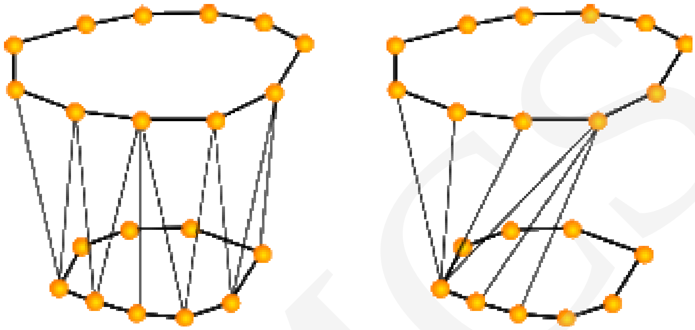

Fig. 5. Contour stitching – tailing. On the left side correct connections; on the right side causing bending of surface

When there are different numbers of contours in adjoining cross-sections, the branching problem arises. It is necessary to determine how the contours should be connected. Some examples of possible connections are depicted in fig. 6. There are methods which calculate the resemblance or overlapping factor.

Most methods try to solve at least one of the described problems whose broad review can be found in the following articles [16,17,18]. But the work by Bajaj et al. [19] and Parker & Attia [15] describe methods completely managing all these problems. Furthermore, it is worth mentioning that not all algorithms create the mesh by joining the contour in pairs, there are solutions [20] which first try to find a global mesh common for the whole object and later fill the slots.

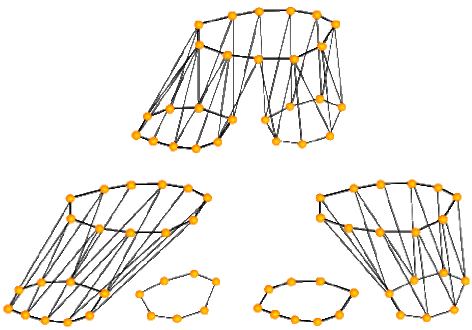

Fig. 6. Contour stitching – branching. On the top the correct connection of contours, while in the bottom line wrong solutions

3.5. Volumetric methods

The volumetric methods (named also field-based or Eulerian methods) define the problem differently [16,18]. In this case from the input data (which is given as a volumetric data) the surface is evaluated and later digitized to the irregular polygonal mesh. For each contour on all cross-sections, the distance function is specified, which with the contour data is transformed to volumetric representation of the data and then the object surface is obtained. Additionally, there are introduced methods [18,21] which utilize the morphing function to achieve smooth surface or divide the object into trapezoids to speed up the performance [22].

3.6. Sweeping

The idea of sweeping [8,23,24,25,26] is to move a closed contour along a given trajectory in the space which results in the object surface definition. Moving the contour might change its shape generated by rotation or scaling. Firstly, this method was utilized only for shapes which do not cut themselves, but the research by Abdel-Malek et al. [22] solved this problem.

4. Methods comparison

4.1. Object appearance

As it was mentioned in the previous section the estimation of the object appearance is subjective. However, two cases might be considered. The first concentrates on the general beauty of the reconstructed object, where the viewer expects smooth surfaces with good resolution. On the other hand, the accuracy plays the important role, where more important are details.

For general appearance it is not necessary to exactly reconstruct the object of interest, therefore the methods based on approximation should give a better result. Moreover, its performance improved by other functionality is a good solution. Hence, for such system demands good results are obtained by: marching cubes, volumetric or sweeping methods (see the first row, Table 1).

When the visualization should resemble the original object and the additional possible changes introduced by approximation are unwanted the interpolation methods should be the solution. This aspect is very important for doctors who cannot examine the medical data loaded with additional errors. Here the resulting polygonal mesh is built only on the vertices given in the input data, therefore the surface sometimes lacks in smoothness. However, it is worth noticing that in the case of medical data, the noise ratio is assumed to be very low, therefore the reconstructed surface resembles the original data. The suggested solutions are: marching voxels, Delaunay triangulation or contour stitching (see the first row, Table 1).

Table 1. Comparison of the object reconstruction algorithms

<table>
<tr><th></th><th>Marching cubes</th><th>Marching vertices</th><th>Delaunay triangulation</th><th>Contour Stitching</th><th>Volumetric methods</th><th>Sweeping methods</th></tr>
<tr><td>Object appearance (reconstruction method)</td><td>approximation</td><td>interpolation</td><td>interpolation</td><td>interpolation</td><td>approximation</td><td>approximation</td></tr>
<tr><td>Resulting surface</td><td>irregular polygonal mesh (triangles)</td><td>irregular polygonal mesh (triangles)</td><td>irregular polygonal mesh (triangles)</td><td>irregular polygonal mesh (quadrilaterals and triangles)</td><td>Points which needs next method for mesh creation</td><td>irregular polygonal mesh (quadrilaterals and triangles)</td></tr>
<tr><td>Opened objects</td><td>No</td><td>No</td><td>No</td><td>Yes</td><td>No</td><td>Yes</td></tr>
<tr><td>Closed objects</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Branching</td><td>Yes</td><td>No</td><td>No</td><td>Yes</td><td>Depends on the method used for triangulation</td><td>No</td></tr>
<tr><td>Holes</td><td>Yes</td><td>No</td><td>No</td><td>Yes</td><td>No</td><td>No</td></tr>
<tr><td rowspan="3">Time</td><td colspan="5">Vertices quantity</td><td>Steps number</td></tr>
<tr><td>Cube resolution</td><td></td><td></td><td></td><td>Additional time for triangulation</td><td></td></tr>
<tr><td colspan="5">Possible parallel implementation</td><td>Not easy</td></tr>
</table>

4.2. Surface representation

The assumption made at the beginning is that the surface should be represented as irregular polygonal mesh. Generally, all presented methods result in this representation, however, in the case of volumetric algorithms it is a second step of its work, which has influences on the processing time (see the second row, Table 1).

4.3. Object features

During the object reconstruction it is important to find a method which is able to retrieve each shape from infinity of shapes that is very difficult to achieve. For instance, most methods which manage well closed objects have problems to work correctly with open ones as those cases are contradictory. For the methods which analyse the point cloud (or volumetric data) without additional information, it is difficult to guess where openings should be. On the other hand, those which use contours as an input, if specified, may or may not close the surface of the contours on the first and last cross-sections. See the third and fourth rows in Table 1 for the gathered results.

The next problem concerns branching structures. For some algorithms this problem was not studied as it takes place in the case of sweeping. For others, like contour stitching, it is one of the problems whose solution was widely searched. For the remaining ones it does not create any problems. For comparison see the fifth row, Table 1.

Finally, the algorithm should not cause additional holes in the surface. It may take place in some methods when their triangulation is specified insufficientlyas in the case of the marching cubes algorithm, when there are not enough triangulation classes or in the contour stitching in the case of branching. The comparison is in the sixth row, Table 1.

4.3. Time

The evaluation time needed by the method to achieve a reconstruction of a given object is crucial for the systems working in the real-time or soft real-time environments. However, the soft real-time reality assumes that it might be necessary to wait a little for the results. This premise is valid and safe for many systems except, for instance object visualization in the computer based surgery. Yet in the case of medical diagnosis, which is the main concern of this article, it is sufficient.

On the other hand, it is worth paying attention to other features which influence the method processing time. As it is easy to notice in every case, the time depends on the input data quantity, as most methods look through each vertex for those working with volumetric data or check every pixel from point cloud. In the case of the marching cubes, the higher resolution, the longer processing time. Moreover, for the volumetric methods there is also a mark-up caused by the need to use an additional method for triangulation. Nevertheless, it can be seen that most of the presented algorithms (except the sweeping method) can be easily implemented for parallel processing which should improve the performance. For comparison see the last row, Table 1.

Conclusions

This article presents the static methods exploited for object reconstruction from point cloud. It includes a brief description of each method and a comparison of their performance in respect to achieved object appearance, impact of noisy data, possible types of object reconstruction (like closed, opened or branching surfaces, etc.) and time consumption. The aim of this comparison is to find which of the presented methods are promising for object reconstruction needed for medical diagnosis. From the data presented in Table 1, it is obvious that the method which fulfils all described requirement is the contour stitching as it works well with the contour data obtained from the scanner, allows creating any shape when problems of correspondence, tailing and branching are well solved and provide reasonable processing time. However, the further research for its development should be carried out, as there are still problems to generate the branching objects correctly without twisting the surface and not creating holes.

References

[1] Edelsbrunner, H., Mücke, E. P. *Three-dimensional Alpha Shapes*. In ACM Transactions of Graphics, (1994).

[2] Lorensen, W. E., Cline, H. E., *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. In Proc. of 14thConference on Computer Graphics and Interactive Technologies, (1987).

[3] Bourke, P., 1994. Polygonising a Scalar Field.
In http://local.wasp.uwa.edu.au/~pbourke/geometry/polygonise.

[4] Bourke, P., *Polygonising a Scalar Field Using Tetrahedrons* In http://local.wasp.uwa.edu.au/ ~pbourke/geometry/polygonise/, (1997).

[5] Chernyaev, E.V., *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*. In Technical Report CERN CN 95-17. CERN, (1995).

[6] Jin J., Wang Q., Shen Y., Hao J., *An Improved Marching Cubes Method for Surface Reconstruction of Volume Data*. In Proc. of World Congress on Intelligent Control and Automation, (2006).

[7] Klein R., Schilling A., Strasser W., *Reconstruction and Simplification of Surfaces from Contours*. In Proc. of 7th Pacific Conference on Computer Graphics and Applications, (1999).

[8] Schmidt, R., Wyvill, B., *Generalized Sweep Templates for Implicit Modeling*. In Proc. of 3rd International Conference on Computer Graphics and Interactive Techniques, (2005).

[9] Lin, Ch.-F., Yang, D.-L., Chung, Y.-Ch., *A Marching Voxels Method for Surface Rendering of Volume Data*. In Proc. of International Conference on Computer Graphics, (2001).

[10] Delaunay, B., *Sur la sphére vide*. In Otdelenie Matematicheskikh i Estestvennykh Nauk, (1934).

[11] Yuan, J.S., Fitzimons, C.J., *A Mesh Generator for Tetrahedral Elements Using Delaunay Triangulation*. In IEEE Transaction on Magnetics, (1993).

[12] Lee, C.-Y., Antonsson, E. K., *Surface Reconstruction of Etched Contours*. In Proc. of Modeling and Simulation of Microsystems, (1999).

[13] Si, H., *TetGen – A Quality Tetrahedral Mesh Generator and Three-dimensional Dalaunay Triangulator*. In http://tetgen.berlios.de/ , (2006).

[14] Ganapathy, S., Dennehy, T. G., *A New General Triangulation Method for Planar Contours*. In Computer Graphics, (1982).

[15] Parker, J. R., Attia, E. N., *Object Reconstruction from Slices for Vision*. In Proc. of International Computer Graphics, (1999).

[16] Braude, I., Marker, J., Museth, K., Nissanov, J., Breen, D., *Contour-Based Surface Reconstruction Using MPU Implicit Models*. In Graphical Models, (2007).

[17] Li, W., Hu, Sh., Zhao, G., *Manufacturable Surface Reconstruction from Complex Contours*. In Emerald Group Publishing Limited, Rapid Prototype Journal, (2004).

[18] Nilsson, O., Breen D., Museth K., *Surface Reconstruction Via Contour Metamorphosis: an Eulerian Approach With Lagrangian Particle Tracking*. In Proc. of IEEE Visualization, (2005).
[19] Bajaj, Ch. L., Coyle, E. J., Lin K.-N., *Arbitrary Topology Shape Reconstruction from Planar Cross Sections*. In Graphical Models and Image Processing. Academic Press, (1996).
[20] Barequet, G., Shapiro, D., Tal, A., *History Consideration in Reconstructing Polyhedral Surfaces from Parallel Slices*. In Proc. of Conference on Visualization, (1996).
[21] Cohen-Or, D., Levin, D., Solomovici, A., *Contour Blending Using Warp-Guided Distance Field Interpolation*. In Proc. of the 7th IEEE Visualization Conference, (1996).
[22] Galin, E., Akkouche, S., *Fast Surface Reconstruction from Contours Using Implicit Surfaces*. In Implicit Surface Conference, (1998).
[23] Abdel-Malek, K., Blackmore, D., Joy, K., *Swept Volumes: Foundations, Perspectives, and Applications*. In International Journal of Shape Modeling, (2000).
[24] Poston, T., Fang, Sh., Lawton, W., *Computing and Approximating Sweeping Surfaces Based on Rotation Minimizing Frames*. In Proc. of 4th International Conference on CAD/CG, (1995).
[25] Schmidt, R., Wyvill, B., *Implicit Sweep Surfaces*. In Technical Report 2005-778-09, Department of Computer Science, University of Calgary, (2005).
[25] Shen, H.-W., Johnson, Ch. R., *Sweeping Simplices: A Fast Iso-Surface Extraction Algorithm for Unstructured Grids*. In Proc. of Conference on Visualization, (1995).